



## 전공 알아보기

### 컴퓨터공학 : 삶의 모습을 향상시키는 기술

#### 컴퓨터공학이란?

컴퓨터공학과는 컴퓨터에 대한 전반적인 모든 것들을 배우는 학과입니다. 컴퓨터는 모니터, 키보드, 본체등을 담당하는 하드웨어와 인터넷, 페이스북, 롤등을 담당하는 소프트웨어로 크게 나눌 수 있습니다. 컴퓨터의 다양한 하드웨어들과, 그 장치들로 돌아가는 수 많은 소프트웨어들은 떼려야 뗄 수 없는 관계입니다. 서로 밀접하게 연관되어있기 때문에 컴퓨터공학과에선 어느 하나만 좋다고 하나만 배우는 것이 아니라, 둘 다 깊게 배우게 됩니다.

#### 소프트웨어와 하드웨어

소프트웨어란, 어떤 기계적인 장치가 있어 직접 볼 수 있는 것이 아닌 컴퓨터 안에만 존재하는 프로그램들을 말합니다. 숫자를 넣으면 계산을 해주는 계산기부터 카카오톡이나 인스타그램 같은 프로그램들, 컴퓨터 자체가 돌아갈 수 있게 해주는 프로그램들이 전부 소프트웨어입니다. 대학에서는 소프트웨어공학, HCL, 기계학습 등의 과목을 통해 소프트웨어를 익힙니다.

하드웨어란, 컴퓨터를 구성하는 것들 중에서 손에 잡히는, 기계적인 장치들을 의미합니다. 작게는 CPU, RAM부터 시작해서 그래픽카드, SSD, 크게는 모니터와 키보드까지도 해당합니다. 대학에서는 컴퓨터 구조, 운영체제 등의 과목을 통해 하드웨어를 익힙니다. 컴퓨터가 작동 중일 때 안의 작은 부품들은 어떤 원리로 동작하고 무슨 신호를 주고받는지, 어떻게 하면 효율적으로 작동하게 만들 수 있는지에 대해 자세히 배웁니다.

마지막으로 프로그래밍이란, 우리가 흔히 말하는 코딩을 의미합니다. 직접 컴퓨터에게 명령을 내려서 원하는 기능을 하게 만듭니다. 컴퓨터는 전기신호 0과 1로만 소통하기 때문에 사람들은 일반적인 영어나 한국어가 아닌 미리 정해진 프로그래밍 언어를 사용하여 코드를 작성하여 컴퓨터와 소통을 합니다. 프로그래밍 언어에는 C, C++, 파이썬, 자바등이 있으며, 특히 파이썬은 가장 입문하기 좋은 프로그래밍 언어입니다.

## 🔍 컴퓨터공학과와 향후 진로

컴퓨터공학과를 졸업한 후 진출할 수 있는 분야는 무수히 많습니다. 먼저, 하드웨어 개발자는 하드웨어와 주변기기에 대한 지식을 바탕으로 하는 직업으로 컴퓨터 하드웨어 시스템을 설계하고, 실제 생산에 필요한 기술들 예를 들어 서버, 네트워크, 시스템 등을 지원하고 개발하는 일을 합니다.

소프트웨어 개발자는 컴퓨터를 부팅하고 사용하게 되는 모든 프로그램을 설계하고 개발하는 일을 합니다. 일반 사용자들이 좀 더 쉽게 컴퓨터를 사용하게 하기 위해서 많은 고민을 합니다. 소프트웨어 개발자는 다양한 분야에서 일을 합니다. 먼저, 웹 프로그래머는 컴퓨터로 사용하는 웹, 예를 들면 네이버나 구글 등 웹 사이트상에서 게시판, 카운터, 회원 가입 및 인증 등을 가능하도록 만들기도 하며, 빠르게 동작 할 수 있도록 최적화 시키기도 하는 등 웹에 관련된 업무를 합니다.

다음으로, 네트워크 프로그래머는 이렇게 만들어진 사이트에 영향을 미치는 네트워크의 성능을 모니터링하며 컨트롤하기 위해 논리적인 네트워크 시스템을 계획 및 설계하여 적용합니다. 또 다른 직업으로는 컴퓨터 보안 전문가가 있습니다. 화이트 해커가 이에 해당됩니다. 컴퓨터 보안 전문가도 있습니다. 정보화 사회에 들어서면서 개인정보 및 여러 중요한 정보들이 생겨났는데 이를 보호하기 위해서 보안정책을 수립하기도 하며, 시스템이 해킹 당하는 것을 미리 방지하고, 해킹 당했을 시 신속히 대응하여 복구하는 업무를 합니다. 그 외에도, 인공지능 전문가, 게임 개발자 등 다양한 분야의 개발자들이 존재합니다.

## 🔍 컴퓨터공학과와 향후 진로

현재 우리가 사는 세상에서는 사실상 사람이 수동으로 하는 것은 점점 없어져가고 있습니다. 많은 작업들이 자동화되고, 상상만 했던 자율주행, 인공지능, 가상현실 같은 신기술들, 암호화폐, 5G 등 최첨단 기술들이 점점 현실화되고 있습니다. 이런 것들을 살펴보면 어느 하나 컴퓨터공학과 관련되지 않은 게 없고, 범위를 더 넓혀보면 물리, 화학, 생물, 신소재 등과 같은 다른 분야와의 협업에도 컴퓨터공학은 빠지지 않습니다. 그만큼 연구던, 기업이던, 창업이던 필요와 수요가 엄청나고, 항상 새로운 기술들, 새로운 트렌드를 가장 먼저 받아들이고 활용할 수 있는 그런 전공입니다.

또한 자신이 원하는 것을 직접 구현할 수 있습니다. 코딩 언어를 배우게 된다면 정말 간단한 사칙연산 계산기부터 시작해서, 테트리스 같은 게임을 만들어 볼 수도 있고 서버를 직접 구축하거나 해킹해보고, 원하는 기능을 가진 모바일 앱을 만들어 볼 수도 있습니다. 간단한 내용은 짧은 기간내에 쉽게 배울 수 있고, 배우면 배울수록 구현할

수 있는 범위가 늘어납니다. 단순히 책만 보고 수업만 듣는 그런 과목이 아닌, 직접 자신만의 코드를 짜고 그 결과물을 눈으로 보고 활용할 수 있다는 부분이 정말 매력적입니다.

## 체험하기

### 컴퓨터와의 첫 대화

※ 파이썬 소개 및 간단한 코딩 배우기!

컴퓨터공학과에 오면 프로그래밍 언어를 배우서 직접 코딩을 통해 원하는 기능을 구현한다고 했죠? 그래서 이번 시간엔 쉽고 다양하게 사용되는 프로그래밍 언어 중 하나인 파이썬이 무엇인지 배우고 직접 파이썬을 이용하여 코딩 하는 방법을 알아보도록 합시다!

파이썬(Python)은 현재 가장 유명한 프로그래밍 언어 중 하나입니다. 1989년 개발되어 현재는 파이썬3까지 나온 상태로 코드를 짜는 규칙, 즉 코딩의 문법이 매우 쉬워서 배우기도 정말 쉬워 초보자가 처음 프로그래밍을 배울 때 추천하고, 각종 수학, 과학 과목들과 인공지능을 연구하기 적합해서 실제로 UNIST에서도 입학하고 1학년 때 파이썬을 배우게 됩니다.

사실 프로그래밍 언어는 거창하게 설명하는 것보다 직접 해보는 것이 훨씬 재미있고 이해도 잘되죠. 바로 알아보러 갑시다! 오늘 수업의 목표는 가상의 음료수 자판기를 만들어보는 것입니다. 동전을 넣고 버튼을 누르면 원하는 음료수가 나오는 그런 자판기 말입니다. 먼저 자판기를 만들기 위해 필요한 것이 무엇인지 생각해 보면, 돈을 넣을 수 있어야 하고 그 돈으로 음료수를 구매해서 음료수와 거스름돈을 내보낼 수 있어야 합니다. 이 기능들을 구현하기 위해서 필요한 것이 무엇인지 차근차근 배워보도록 합시다.

#### 1. 변수 저장 및 사칙연산

컴퓨터가 값을 저장하는 곳을 '변수' 라고 합니다. 말 그대로 변할 수 있는 수로써 이러한 변수에 값을 저장해서 사용하게 됩니다. 컴퓨터 세계에서 등호 '=' 는 수학의 '같다' 를 의미하는 등호와 는 조금 다르게 사용됩니다. 컴퓨터는 등호를 '오른쪽에 있는 값을

```

1 school = "유니스트"
2 school

'유니스트'

1 a = 2 ; b = 4
2 c = a+b ; print(c)
3 c = a-b ; print(c)
4 c = a*b ; print(c)
5 c = a/b ; print(c)

6
-2
8
0.5

```

왼쪽으로 넣는다' 라는 명령으로 인식합니다. 즉, 컴퓨터는 school = 유니스트 라고 입력하면 school이라는 변수에 유니스트 라는 내용을 저장하고, age = 21 이라고 입력하면 age라는 변수에 21 이라는 내용을 저장하게 됩니다. 변수들끼리 계산한 결과를 다른 변수에 저장할 수도 있습니다. a=2, b=4 라고 해봅시다. C = a+b, c = a-b, c = a\*d, c = a/b를 하면 c에 각각 사칙연산을 한 결과인 6, -2, 8, 0.5가 저장됩니다.

## 2. 입출력

변수에 값을 저장하는 방법을 배웠으니 이번에는 값을 입력 받는 방법과 결과를 출력하는 방법을 알아보시다. 입력과 출력은 input() 과 print()를 이용해서 이루어지게 됩니다. a=input() 을 하면 키보드로 입력하는 값이 그대로 a에 들어가게 되고, 반대로 a를 출력하고 싶으면 print(a) 하시면 됩니다.

## 3. 제어문 조건문

여러분 “if” 가 무슨 뜻인지 아시나요? “만약에”라는 뜻입니다. 이렇게 어떤 특정한 상황에 맞춰서 만약에 A 상황에서는 a를 수행하고 B상황에서는 b를 수행 할 수 있습니다.

예를 들어 볼까요? 자판기에서 음료를 뽑는데 만약에 사이다를 눌렀으면 사이다가 나와야하고 콜라를 누르면 콜라가 나와야 합니다.

이 문장을 다시 말하면 if 사이다: 사이다를 출력 Else if 콜라 : 콜라를 출력 예시를 볼까요?

```
1 Number = 5
2
3 if Number < 10: print("10보다 작은 숫자입니다.")
4 else: print("10보다 큼니다")
5
```

10보다 작은 숫자입니다.

위의 예에서 값이 5 인 'Number'이라는 변수를 선언하고 if 문에서 숫자가 10보다 작은 지 여부를 확인하고 조건이 참이면 if 블록 내부의 문 명령문이 실행되고 아니면 else 내부의 명령문이 실행됩니다.

이렇게 yes or no로 두가지로 나눌 수 있을때도 있지만 자판기처럼 여러가지의 조건이 있을 때는 elif 를 사용합니다. elif 는 else-if의 줄인 말입니다. 바로 예제를 살펴볼까요?

```

1 음료 = "콜라"
2 if 음료 == "사이다": print("사이다를 선택하였습니다.")
3 elif 음료 == "콜라": print("콜라를 선택하였습니다.")
4 elif 음료 == "아침햇살": print("아침햇살을 선택하였습니다.")
5 elif 음료 == "이프로": print("이프로를 선택하였습니다.")
6 elif 음료 == "웅달생물": print("웅달생물을 선택하였습니다.")
7 else : print("잘못 입력되었습니다. 다시 선택하여 주세요")

```

콜라를 선택하였습니다.

위의 예에서 값이 콜라 인 음료이라는 변수를 선언하고 여러가지 조건문을 두어 음료 가 가지고 있는 값이 무엇인지 알려줍니다.

이렇게 프로그램은 실행 흐름을 제어 해야 합니다. 주어진 조건이 알맞을 경우에만 명령문을 실행하고 만족하지 않으면 다른 명령문을 실행합니다. 이를 제어문 또는 조건문이라고 합니다.

#### 4. 반복문

여러분 만약 1부터 10까지 출력하고 싶으면 어떻게 할까요?

아마 print(1), print(2) ... print(10)을 하겠죠? 하지만 만약에 100까지 출력해야 한다면? 아니면 10000까지 출력해야 한다면 어떻게 할까요? 이럴 때 쓰이는 것이 반복문입니다.

반복문에는 for문과 while문이 있습니다. For문 먼저 살펴보면, 다음과 같이 사용합니다.

for [변수] in 숫자들, 리스트, :

[수행부분]

숫자들이 하나씩 변수에 들어가서 수행하게 됩니다! 예제로 확인하여 봅시다!

```

for i in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
    print(i)

```

i라는 변수에 0부터 10까지 순서대로 들어간 뒤, print(i) 문장이 수행하게 됩니다.

다음으로는 while문입니다. While문도 for문처럼 “반복”할 때 사용하지만 무작정 반복하는 것이 아니라 조건에 맞을때 반복하게 됩니다. For문과 if문이 합쳐진 모양입니다.

while [조건문]:

[수행부분]

while 반복문은 [조건문]이 참(True)인 경우 내부의 수행 부분을 진행하고, [조건문]이 거짓(False)인 경우 while문을 빠져나갑니다. 즉 조건문이 참일 동안(while)반복을 진행하는 겁니다!

"열 번 찍어 안 넘어가는 나무 없다"는 속담을 while문을 이용해 보자면 다음과 같습니다.

```
나무찍은횟수 = 0
while 나무찍은횟수 < 10:
    나무찍은횟수 = 나무찍은횟수 +1
    print("나무를 %d번 찍었습니다." %나무찍은횟수)
    if 나무찍은횟수 == 10:
        ("나무 넘어갑니다.")
```

저희 아웃풋을 한번 예상해볼까요?

## 5. 자판기 실습

그럼 배운 내용을 기반으로 자판기를 만들어 봅시다. 아래의 이미지는 자판기의 소스코드, 즉 원본 파일입니다. 처음부터 차근차근 배워봅시다!

```
1 money = int(input("돈을 넣어주세요: "))
2
3 while True:
4     if money < 500:
5         print("더 이상 음료를 살 수 있는 잔액이 아닙니다. 자판기를 종료합니다.")
6         break
7
8     choice = int(input("음료를 선택해주세요: "))
9
10    if choice == 1:
11        if money >= 500:
12            money = money - 500
13            print("주스를 고르셨습니다. 가격은 500원 입니다. 잔액은 ", money, "원 입니다.", sep="")
14        else:
15            print("주스를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
16    elif choice == 2:
17        if money >= 1000:
18            money = money - 1000
19            print("콜라를 고르셨습니다. 가격은 1000원 입니다. 잔액은 ", money, "원 입니다.", sep="")
20        else:
21            print("콜라를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
22    elif choice == 3:
23        if money >= 1500:
24            money = money - 1500
25            print("커피를 고르셨습니다. 가격은 1500원 입니다. 잔액은 ", money, "원 입니다.", sep="")
26        else:
27            print("커피를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
```

```

28 elif choice == 4:
29     if money >= 2000:
30         money = money - 2000
31         print("우유를 고르셨습니다. 가격은 2000원 입니다. 잔액은 ", money, "원 입니다.", sep="")
32     else:
33         print("우유를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
34 else:
35     print("없는 선택지를 고르셨습니다. 다른 번호로 다시 선택해주세요.")

```

돈을 넣어주세요: 2000  
음료를 선택해주세요: 1  
주스를 고르셨습니다. 가격은 500원 입니다. 잔액은 1500원 입니다.  
음료를 선택해주세요: 2  
콜라를 고르셨습니다. 가격은 1000원 입니다. 잔액은 500원 입니다.  
음료를 선택해주세요: 1  
주스를 고르셨습니다. 가격은 500원 입니다. 잔액은 0원 입니다.  
더 이상 음료를 살 수 있는 잔액이 아닙니다. 자판기를 종료합니다.

```

1 money = int(input("돈을 넣어주세요: "))

```

우선 자판기라면 돈을 입력받아야겠죠? 앞에서 배운 입력함수인 input을 이용해서 돈을 받은 후, money라는 변수에 저장합니다.

```

3 while True:
4     if money < 500:
5         print("더 이상 음료를 살 수 있는 잔액이 아닙니다. 자판기를 종료합니다.")
6         break
7
8     choice = int(input("음료를 선택해주세요: "))
9

```

그리고 while에 True라는, 항상 참인 값을 넣어서 우리가 돈이 없을 때까지 자판기가 입력을 받을 수 있도록 합니다. 또한, 만약 돈이 500원 미만으로 떨어진다면, 입력을 그만 받을 수 있도록 조건문을 사용합니다.

만약 돈이 충분하다면, 어떠한 음료수를 먹을 것인지 고를 수 있도록 입력값을 받도록 합니다.

```

10 if choice == 1:
11     if money >= 500:
12         money = money - 500
13         print("주스를 고르셨습니다. 가격은 500원 입니다. 잔액은 ", money, "원 입니다.", sep="")
14     else:
15         print("주스를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
16 elif choice == 2:
17     if money >= 1000:
18         money = money - 1000
19         print("콜라를 고르셨습니다. 가격은 1000원 입니다. 잔액은 ", money, "원 입니다.", sep="")
20     else:

```

```

21     print("콜라를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
22 elif choice == 3:
23     if money >= 1500:
24         money = money - 1500
25         print("커피를 고르셨습니다. 가격은 1000원 입니다. 잔액은 ", money, "원 입니다.", sep="")
26     else:
27         print("커피를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
28 elif choice == 4:
29     if money >= 2000:
30         money = money - 2000
31         print("우유를 고르셨습니다. 가격은 1000원 입니다. 잔액은 ", money, "원 입니다.", sep="")
32     else:
33         print("우유를 구입하기 위한 돈이 부족합니다. 다른 음료를 선택해주세요.")
34     else:
35         print("없는 선택지를 고르셨습니다. 다른 번호로 다시 선택해주세요.")

```

이번에 구현한 자판기에서는 총 선택지가 4가지가 있습니다. 주스, 콜라, 커피 그리고 우유가 있습니다. 각 선택지의 가격은 500원, 1000원, 1500원 그리고 2000원 입니다. 각 선택지에 맞는 숫자를 입력하면, 우선 돈이 충분한지 확인하게 됩니다. 만약 돈이 충분하지 않다면, 처음으로 다시 돌아가서 다른 선택지를 고르게 됩니다. 만약 돈이 충분하다면 가격에 맞는 돈을 잔액에서 빼가게 됩니다.

```

돈을 넣어주세요: 2000
음료를 선택해주세요: 1
주스를 고르셨습니다. 가격은 500원 입니다. 잔액은 1500원 입니다.
음료를 선택해주세요: 2
콜라를 고르셨습니다. 가격은 1000원 입니다. 잔액은 500원 입니다.
음료를 선택해주세요: 1
주스를 고르셨습니다. 가격은 500원 입니다. 잔액은 0원 입니다.
더 이상 음료를 살 수 있는 잔액이 아닙니다. 자판기를 종료합니다.

```

위는 실제로 저희가 만든 자판기 코드를 돌려본 결과값입니다. 여러분도 따라해 보세요!

## 🔍 QUIZ

Q1. 직접 컴퓨터에 명령하여 원하는 기능을 수행하도록 만들 수 있는 것은 무엇일까요?

Q2. D라는변수에 7X123의 값을 저장하는 코드를 직접 작성해보자.

Q3. "4. 반복문"에서 마지막에 나오는 코드의 결과값은 무엇일까요?

Q4. "5. 자판기 실습"에서 순서대로 3000, 2, 3, 4, 1을 입력하면 어떠한 결과값이 나올까요?